

2 Feb Assignment

February 6, 2023

0.1 PYTHON

0.1.1 Assignment Questions

0.1.2 Q1

[2]: *# Q1. What are the characteristics of the tuples? Is tuple immutable?*

[3]: *# Ans:- These tuples are an ordered collection of elements of different data types. Furthermore, we represent them by writing the elements inside the parenthesis separated by commas. We can also define tuples as lists that we cannot change. Therefore, we can call them immutable tuples.*

0.1.3 Q2

[4]: *# Q2. What are the two tuple methods in python? Give an example of each method. Give a reason why tuples have only two in-built methods as compared to Lists.*

[5]: *# Ans:- In Python, tuples are immutables. Meaning, you cannot change items of a tuple once it is assigned. There are only two tuple methods count() and index() that a tuple object can call.*
The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable. This means that tuples cannot be changed while the lists can be modified.
Tuples are more memory efficient than the lists.
When it comes to the time efficiency, tuples have a slight advantage over the lists especially when we consider lookup value.
If you have data that shouldn't change, you should choose tuple data type over lists.

0.1.4 Q3

[6]: *# Q3. Which collection datatypes in python do not allow duplicate items? Write a code using a set to remove duplicates from the given list.*

[7]: *# Ans:- 31.2 Python Collection Types*

```
# Sets Sets are a collection that is unordered and unindexed. They are mutable,
↳(changeable) but do not allow duplicate values to be held.
```

```
[13]: List = [1, 1, 1, 2, 1, 3, 1, 4, 2, 1, 2, 2, 2, 3, 2, 4, 3, 1, 3, 2, 3, 3, 3, 4,
↳4, 1, 4, 2, 4, 3, 4, 4]
```

```
[14]: my_final_List = set(List)
```

```
[15]: print(list(my_final_List))
```

```
[1, 2, 3, 4]
```

0.1.5 Q4

```
[16]: # Q4. Explain the difference between the union() and update() methods for a set.
↳Give an example of each method.
```

```
[17]: # Ans. pdate() adds all missing elements to the set on which it is called,
↳whereas set. union() creates a new set. Consequently, the return value of
↳set.
```

```
[18]: # Union of Sets:
# In set theory, the union of two or more sets is the set which contains all,
↳the elements ( distinct ) present in all the sets. For example :
```

```
[19]: A = {1, 2, 3, 4, 5}
B = {4, 5, 6, 7, 8, 9}
```

```
[20]: print('A : ',A)
```

```
A : {1, 2, 3, 4, 5}
```

```
[21]: print('B : ',B)
```

```
B : {4, 5, 6, 7, 8, 9}
```

```
[22]: print('A : ',B)
```

```
A : {4, 5, 6, 7, 8, 9}
```

```
[23]: # This method is used to return the union of a set and the set of elements from,
↳one or more iterable like string, list, set. It is very similar to union(),
↳method, with difference is that where union() method create and return a new
↳set, containing all the elements ( distinct ) present in all the iterables,
↳update() method updates the set on which this method is called with all the
↳distinct elements present in all the iterables
```

```
[24]: A = {1, 2, 3, 4, 5}
      B = {4, 5, 6, 7, 8, 9}
```

```
[25]: A |= B
```

```
[26]: print('A Union B : ',A)
```

```
A Union B : {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
[27]: print('B Union A : ',B)
```

```
B Union A : {4, 5, 6, 7, 8, 9}
```

```
[28]: A.update(B)
```

```
[29]: print('A Union B Union : ',A)
```

```
A Union B Union : {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

0.1.6 Q5

```
[30]: # Q5. What is a dictionary? Give an example. Also, state whether a dictionary
      ↪ is ordered or unordered.
```

```
[31]: # Q5. What is a dictionary? Give an example. Also, state whether a dictionary
      ↪ is ordered or unordered.
```

```
[32]: capital_city = {"Nepal": "Kathmandu", "Italy": "Rome", "England": "London"}
```

```
[33]: print(capital_city)
```

```
{'Nepal': 'Kathmandu', 'Italy': 'Rome', 'England': 'London'}
```

```
[34]: numbers = {1: "One", 2: "Two", 3: "Three"}
```

```
[35]: print(numbers)
```

```
{1: 'One', 2: 'Two', 3: 'Three'}
```

0.1.7 Q6

```
[36]: # Q6. Can we create a nested dictionary? If so, please give an example by
      ↪ creating a simple one-level nested dictionary.
```

```
[37]: # Ans:- A nested Python Dictionary is a dictionary that has many Dictionaries.
      ↪ We can create a dictionary with that contain n number of dictionaries
```

```
[38]: nested_dict = { 'dict1': {'key_A': 'value_A'},  
                      'dict2': {'key_B': 'value_B'}}
```

```
[39]: nested_dict
```

```
[39]: {'dict1': {'key_A': 'value_A'}, 'dict2': {'key_B': 'value_B'}}
```

```
[40]: dict1 = {'language' : 'Python', 'course': 'Data Science Masters'}
```

```
[41]: dict1.setdefault('topics', ['Python', 'Machine Learning', 'Deep Learning'])
```

```
[41]: ['Python', 'Machine Learning', 'Deep Learning']
```

```
[42]: print(dict1)
```

```
{'language': 'Python', 'course': 'Data Science Masters', 'topics': ['Python',  
'Machine Learning', 'Deep Learning']}
```

0.1.8 Q7

```
[43]: #Q7. Using setdefault() method, create key named topics in the given dictionary  
      ↪and also add the value of  
      # the key as this list ['Python', 'Machine Learning', 'Deep Learning']  
      # dict1 = {'language' : 'Python', 'course': 'Data Science Masters'}
```

```
[44]: dict1 = {'language' : 'Python', 'course': 'Data Science Masters'}
```

```
[45]: dict1.setdefault('topics', ['Python', 'Machine Learning', 'Deep Learning'])
```

```
[45]: ['Python', 'Machine Learning', 'Deep Learning']
```

```
[46]: print(dict1)
```

```
{'language': 'Python', 'course': 'Data Science Masters', 'topics': ['Python',  
'Machine Learning', 'Deep Learning']}
```

0.1.9 Q8

```
[47]: # Q8. What are the three view objects in dictionaries? Use the three in-built  
      ↪methods in python to display  
      # these three view objects for the given dictionary.  
      dict1 = {'Sport': 'Cricket' , 'Teams': ['India', 'Australia', 'England', 'South',  
      ↪Africa', 'Sri Lanka', 'New Zealand']}
```

```
[48]: # The three view objects in dictionaries are keys(), values() and items().
```

```
[49]: #To display keys
print(dict1.keys())
#Output: dict_keys(['Sport', 'Teams'])
```

```
dict_keys(['Sport', 'Teams'])
```

```
[50]: #To display values
print(dict1.values())
#Output: dict_values(['Cricket', ['India', 'Australia', 'England', 'South
↪Africa', 'Sri Lanka', 'New Zealand']])
```

```
dict_values(['Cricket', ['India', 'Australia', 'England', 'South Africa', 'Sri
Lanka', 'New Zealand']])
```

```
[ ]:
```